RMX/80™ INTERACTIVE CONFIGURATION UTILITY USER'S GUIDE

Manual Order Number: 142603-001

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

A178/380/15K FL

PREFACE

The RMX/80 Interactive Configuration Utility (ICU80) User's Guide describes the operation and use of ICU80.

The RMX/80 User's Guide (manual order number: 9800522) gives details in programming user's tasks and describes how to code application tasks. Both manuals must be used to configure an application system under ICU80.

It is assumed that the reader of this manual has some experience with each of the following:

- Either the Intellec Microcomputer Development System or the Intellec series II Microcomputer Development System and their operating system ISIS-11.
- Any RMX/80 supported language.
- Intel's Real-Time, Multitasking Executive (RMX/80).

It is also assumed that the reader has designed a system (using the RMX/80 User's Guide) and is prepared to generate the software configuration.

RELATED PUBLICATIONS

- RMX/80 User's Guide, 9800522.
- ISIS-I1 User's Guide, 9800306.
- BASIC80 Reference Manual, 9800758.
- FORTRANSO Programming Manual, 9800481.
- ISIS-I1 FORTRAN80 Compiler Operator's Manual, 9800480.
- 8080/8085 Assembly Language Reference Manual, 9800301.
- PL/M Programming Manual, 9800268.
- ISIS-I1 PL/M Compiler Operators Manual, 9800300.

CONTENTS

ILLUSTRATIONS

CHAPTER 1
INTRODUCTION

An RMX/80 system is created from the RMX nucleus and some number of segments of application code. These segments of application code are called tasks. Some of these tasks are supplied with the system such as the nucleus and extensions; others are supplied by the user and are dependent on the application. Selecting and describing these tasks is the process of configuration for an RMX/80 system.

The Interactive Configuration Utility (ICU80) for RMX/80 is a tool used by the application/system programmer, during systems development, to configure an RMX/80 based system. This utility is used to automate the configuration, linking, and locating operations described in the RMX/80 User's Guide. ICU80 is used to produce the configuration object module, the controller addressable memory (CAM) module, and a submit file containing the link and locate commands required to build the RMX/80 application system.

By producing the configuration object module in object format, rather than source, ICU80 overcomes some of the restrictions imposed by the manual method described in the RMX/80 User's Guide. ICU80 runs on an Intel Microcomputer Development System under ISIS-I1 and interacts with the user to obtain the parameter values needed to configure the desired system.

This manual describes the operation and use of ICU80. The RMX/80 User's Guide provides details on programming application tasks. Both manuals must be read and understood before attempting to configure a system.

ICU80 will prompt for all required parameters needed to tailor the RMX/80 system to a particular configuration. This is done by supplying, as input to ICU80, a file known as the description file. Initially the default description file, supplied with the system, is used as input. This file contains default values for all system parameters. These values are changed by the operator using the Change command described later.

The final output of ICU80 consists of a set of files. One of the files is a configuration object module for the specified RMX/80 application system. This module is used by the RMX/80 based system to setup its environment. Another file is a CAM module, if one is required. Finally, there is a Submit file containing the commands necessary to complete the configuration process. These files are generated only by user request (see Generate command) and then only if no required parameters are missing. Figure 1-1 shows the relationship of the various files and parts of the ICU80.

USE ENVIRONMENT

ICU80 runs on an Intel Microcomputer Development System, as a program, under ISIS-11, version 3.4 or later. ICU80 requires the Intel Microcomputer Development System to have the following as a minimum:

- 64K of memory
- Two single density or one double density floppy disk drive(s)

  An interactive operators terminal

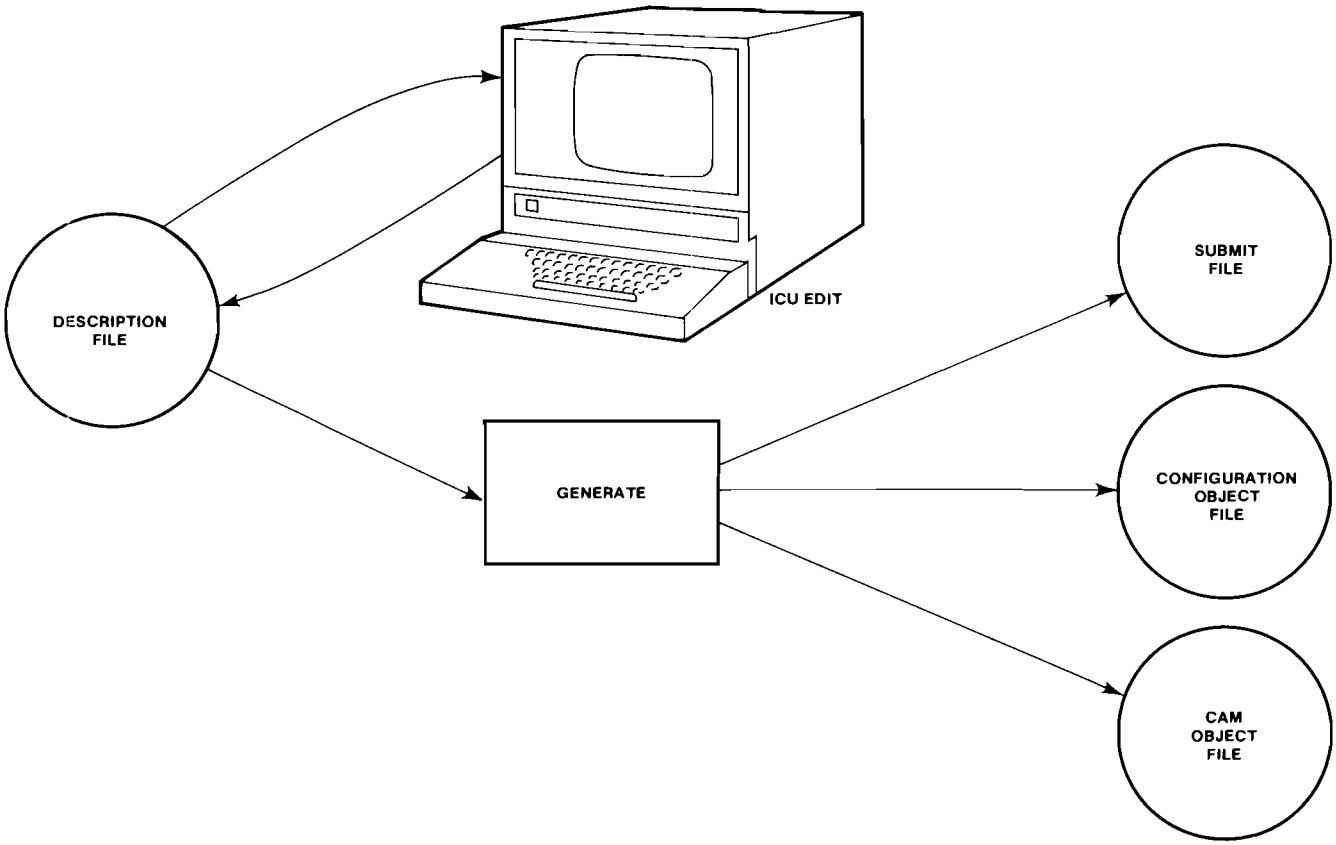

Figure 1-1. ICUSO Interrelationship

CHAPTER 2
BASIC CONCEPTS OF ICU80

INTRODUCTION

This chapter describes the basic concepts of ICU80. It also describes the various commands that can be used by ICU80 and explains the prompts.

OUTPUT FILES

ICU80 is used to produce several files required to configure an RMX/80 system. The first file to be created is the description file. Once the description file has been created the operator may use the Generate command (described in Chapter 3) to produce the configuration object module, CAM module (if required), and the submit file.

DESCRIPTION FILE

The description file contains all the information required to produce the configuration object module, CAM module, and the submit file. The description file is created by the operator in an interactive editing session at the console terminal. The operator supplies information describing the target hardware and software configuration in response to a series of prompts displayed on the console terminal.

The information in the description file provides ICU80 a description of the following:

- Hardware environment.
- Which user tasks and exchanges are to be included in the initial task table and initial exchange table.
- Which RMX/80 extensions are to be included with the application system.

The description file also provides the information necessary to create the submit file. Most of the parameters in the description file are unique entries. That is, they may only appear once in a description file. An example of a unique entry is the parameter CPU TYPE. The value specified for CPU TYPE will vary from system to system, but each system will have only one CPU TYPE specified.

There are, however, some parameters which may be included multiple times. These parameters describe a series of different items of a similar type. For example, each user task to be included in the initial task table must be included in the description file.

The complete description of each task is referred to as a list entry. All of the list entries included in succession are referred to as a list. There are several lists included in the description file. The last item of each list is represented by the primary entry with a default value of ***. This is also referred to as the null list entry.

The parameters which describe each user task consist of a primary entry, TASK NAME, followed by several secondary entries, STACK LENGTH, PRIORITY, etc., which complete the description of the task. Any number of tasks may be included in the description file; therefore, the primary entry TASK NAME is not a unique parameter.

The default description file contains only the null list entry for each list. The operator adds entries to a list by specifying a value, for the primary entry when the null list entry is displayed. The list may be terminated by entering a carriage return in response to the null list entry. For a complete description of the editing commands available, see Chapter 3.

Once the description file has been created, the operator may use the Generate command (described in Chapter 3) to produce the Configuration Object module, CAM module (if required), and the Submit file.

CONFIGURATION OBJECT FILE

The Configuration Object file contains the relocatable object code representing the configuration object module.

CAM OBJECT FILE

The Controller Addressable Memory Module (CAM) file contains the relocatable object code for CAM.

SUBMIT FILE

The Submit file contains the sequence of commands necessary to link and locate the complete RMX/80 application system. This file is submitted by the user at the end of the configuration process.

PROMPTS

ICU80 was designed as an interactive system to aid the user in configuring an application system for

RMX/80. Therefore, each entry is requested by a prompt followed by a colon (:). If a value already exists for that entry (default or previously input) it will be listed following the colon. Some prompts appear only if they are required because of a previous response.

When ICU80 is first initiated, it displays a list of the available commands followed by a request to enter the desired command. At any time the operator does not understand the prompt, an explanation may be requested by typing a "?" followed by a carriage return. ICU80 responds with a few lines of explanation describing the meaning of the prompt and what is expected. The prompt is then repeated. The operator can return to the command prompt by entering an escape character and a C followed by a carriage return in response to any prompt.

The following commands are available (for more detail refer to Chapter 3):

Command

R (replace)
C (change/create)
G (generate)
L (list)
Q (quit)
E (exit)

SPECIAL COMMANDS

To aid the user in the operation of ICU80, the operator is provided with a set of special commands. Their use is explained in detail later in Chapter 3. They are listed here as a summary. The operator may obtain a list of these special commands by entering an escape character (ESC) followed by HELP or H and a carriage return (described in Chapter 3).

The following special commands are available (see Chapter 3 for more details):

KEY	MEANING
(ESC) H	Displays list of special commands.
?	Request explanation.
(ESC) ?	Request explanation.
(ESC) B	Backup to previous line.
(ESC) C	Return to command mode.
(ESC) I	Insert a list element preceding present list element.
(ESC) D	Delete current list element.
(ESC) R	Insert comment preceding present line.
(ESC) F parm	Find the listed parameter and display it.

CONSTANTS

INTEGERS

ICU80 recognizes several varieties of integer constants. As each value is entered, it is checked to see that it is within the allowable range for the given parameter. All integer formats may be used interchangeably. The list command lists constants in the format in which they were entered. The formats that are recognized by ICU80 are as follows:

a. Decimal Data
Each decimal number may be identified by the letter D immediately after its last digit or may stand alone. Any number not specifically identified is assumed to be decimal.
0 to 65535 unsigned.

b. Hexadecimal Numbers
Each hexadecimal number must begin with a numeric digit (0 through 9) and must be followed by the letter H.
0H to OFFFFH unsigned.

c. Octal Data
Each octal number must be followed by the letter O or the letter Q.
0 0 to 177777O unsigned.

d. K-Format
Adding the K to a decimal number is equivalent to multiplying the decimal number by 1024. The K-format is commonly used for describing memory size.
0K to 64K unsigned.

DEVICE/FILE NAMES

Certain parameters require a device/file name. ICU80 recognizes any standard ISIS-I1 device/file format.

IDENTIFIERS

Certain parameters require an identifier. ICU80 recognizes any language compatible string of non-blank characters up to 32.

CHAPTER 3
GENERATING THE CONFIGURATION

INTRODUCTION

Generating the software configuration for the application system consists of several major activities. They are as follows:

a. Defining system components.
b. Initiating ICU80.
c. Preparing the configuration object module.
d. Linking and locating the application system.

DEFINING SYSTEM COMPONENTS

Before the configuration object module can be created or submitted, several questions must be answered:

a. What user tasks are to be included in the Initial Task Table, and what are the characteristics of these tasks (priorities, etc.)?
b. What user exchanges are to be defined when the system is initialized?
c. What RMX/80 extensions are to be included in the system, and what are the characteristics of these extensions (priorities, etc.)?
d. What are the characteristics of each disk controller in the system?
e. What are the characteristics of each disk drive in the system?
f. How are system buffers, for disk files, to be allocated?

INITIATING ICU80

The format of the input for program execution under ISIS-I1 is as follows:

ICU80 (input file TO) Output File

Where:

ICU80 = file name of the Interactive Configuration Utility.

Input File = (Optional) When specified, identifies the description file to be used as input to the configurator. If not specified, the default description file is used.

Output File = (Required) Specifies the device/file name where the modified description file is to be saved.

Rules for file name specification on Initialization

1. If both input and output file names are specificd, the input file must exist and the output file must not exist. Failure to meet these conditions will cause an error message to be displayed and ICU80 to be terminated.

2. If a single file name is specified and the file named does not exist, then the system default description file is used as input, and the new file name is opened for output.

3. If a single file name is specified and the file exists, the file will be backed up and then used as input. The file name specified will then be opened for output.

Example:

ICU80 :F1:SYS1.DSC

The example assumes the ICU80 is contained on drive 0. The operator wishes to create a description file called SYS1.DSC on drive 1. It is assumed, in this example, that the output file (:F1:SYS1.DSC) is a new file.

TEMPORARY FILES

Several temporary files are created and used by ICU80. These files will be placed on the same diskette as the output file that was specified at initiation time. These files will be deleted when ICU80 is exited.

PREPARING THE CONFIGURATION MODULE

COMMAND PROMPT

When ICU80 is first initiated, it displays a list of the available commands followed by a request to enter the desired command. At any time the operator does not understand the prompt, an explanation may be requested by typing a "?" followed by a carriage return. ICU80 responds with a few lines of explanation describing the meaning of the prompt and what is expected. The prompt is then repeated. The operator can return to the command prompt by entering a command character (ESC) and a C followed by a carriage return in response to any prompt.

The following commands are available when ICU80 is in the command mode:

Command	
R	(replace)
C	(change/create)
G	(generate)
L	(list)
Q	(quit)
E	(exit)

REPLACE COMMAND

The Replace command allows the operator to replace the Escape command character (ESC) with any desired character. The default command character is the escape key (ESC).

CHANGE/CREATE COMMAND

To modify the description file, the operator enters the Change/Create command. This places ICU80 in the edit mode. This is accomplished by typing C (or CHANGE) following the prompt, COMMAND.

The format of the Change command is as follows:

COMMAND:
C(HANGE)(parameter(,value(,parameter)))

NOTE

If both parameters are specified, the first parameter entered must be a primary list entry. The second parameter entered must be a secondary list entry.

The Change/Create command causes ICU80 to prompt for each parameter in the description file followed by its present value. The operator responds by entering a new value or carriage return if the present value is acceptable.

If the operator is changing a specific value in the description file, the parameter for the value to be changed can be appended to the Change/Create command. This will cause ICU80 to search the description file for that parameter. If ICU80 finds the parameter, it will display the parameter followed by its current value.

For example, if an operator wanted to change the priority under which the DEBUGGER is to run, the following would be entered.

CHANGE DBGPRI(cr)

Because the debugger priority is a unique parameter (not part of a list) only the parameter need be entered. This results in ICU80 searching the description file for the parameter entered. If ICU80 finds the parameter that was entered, it lists the parameter followed by its current value.

For example, if the search for the parameter was successful, the following would be displayed:

DBGPRI: 50

The operator may now change the priority of the debugger by typing the new priority followed by a carriage return.

If the desired parameter is part of a list, the current value can be appended to the parameter that is entered. If the current value is appended, it will cause ICU80 to search the description file for that parameter and value.

The following are examples of lists:

```
TASK NAME: TSK1
  ENTRY POINT: TSK1
  PRIORITY: 35
  STK LENGTH: 24
  DFLT EXCHG: RQL2EX
TASK NAME: ***

LINK: :F1: CONSL
LINK: :F1: TMEUP
LINK: ***
```

The following example shows the entries necessary to change a particular parameter value in a list.

CHANGE TASK NAME, TSK1, PRIORITY(cr)

This would cause ICU80 to search the description file for the task name TSK1 and display the parameter PRIORITY with its current value. To change the value, the operator enters the new value followed by a carriage return.

If the desired parameter is part of a list, and no value is specified, ICU80 will search the description file for the first parameter that matches. This parameter will be the first parameter entry in the list with this parameter name.

If the operator wishes to add additional entries to a list, the beginning parameter for the desired list is entered followed by three asterisks(***). This causes ICU80 to search the description file for the end of the specified list.

SPECIAL COMMANDS: To aid the user in the operation of ICU80, the operator is provided with a set of special commands. Any time ICU80 is in edit mode, the operator may obtain a list of these special commands by entering a command character (ESC), HELP or H, and a carriage return.

NOTE

If the user's terminal does not have the escape key, it can be changed to any other character by using the REPLACE command.

The list of special commands is as follows:

KEY	MEANING
(ESC) H	Help displays list of special commands.
?	Request explanation.
(ESC) ?	Request explanation.
(ESC) B	Backup to previous line.
(ESC) C	Return to command mode.
(ESC) I	Insert preceding present list element.
(ESC) D	Delete current list element.
(ESC) R	Insert comment preceding present line.
(ESC) F parm	Find the listed parameter and display it.

Two of the special commands can be used any time ICU80 prompts for an input. They are the ? and the (ESC) C commands.

The "?" key allows the operator to request information about prompt that is being displayed. The "?" causes ICU80 to print a message explaining the meaning of the parameter and then the prompt will again be displayed. The operator may then change the parameter value by typing in the new parameter and pressing carriage return, or if the value is acceptable, simply press carriage return.

The carriage return is a termination character. Any time it is entered it terminates the input and ICU80 continues.

The (ESC) B command allows the operator to back-up one prompt from the present prompt line. The previous prompt is displayed and the operator may continue from that point. If an attempt is made to back-up beyond the beginning of the description file, ICU80 will exit Change and return to command mode.

The (ESC) C command allows the operator to return to command mode at any time. When the (ESC) C command is used, the associated parameter is not changed.

The (ESC) I command allows the operator to insert a list entry ahead of the present prompt line if the cursor is positioned at the primary entry of the list. A parameter is displayed and the operator may continue from that point.

For example: In the following list, assume that the cursor is pointing to TASK NAME.

TASK NAME: TSK2 (ESC) I (cr)

This would result in the following

TASK NAME:

The operator enters the new task name and presses carriage return. ICU80 will then prompt for the secondary parameters in the list.

If the list is a single parameter list (e.g. LINK), the insert may be made anywhere in the list.

For example:

LINK: :F1:CONSL
LINK: :F1:TMEUP
LINK: :F1:SCRCH (ESC) I (cr)

This would result in the following prompt being displayed and the prompt and value entered being placed ahead of the :F1:SCRCH value.

LINK:

The (ESC) D command allows the operator to delete a list entry or a comment from the description file. The cursor will back up to the primary parameter for the list entry that the cursor was in and delete all parameters that were contained in that list entry.

The (ESC) R command allows the operator to insert a comment line (or lines) preceding the parameter that is being displayed. (See COMMENTS).

The (ESC) F command allows the operator to search through the description file for a particular parameter.

For example, if an operator wanted to change the priority that TSK2 is to be run under, the following would be typed:

(ESC) F TASK NAME TSK2, PRIORITY (cr)

This would result in ICU80 searching the description file for the parameters entered. If ICU80 finds the parameters that were entered, it will display the parameter followed by its current value.

For example, if the search for the parameters was successful, the following would be displayed:

PRIORITY: *35*

The operator now may change the priority of TSK2 by typing the new priority followed by a carriage return.

The operator may add to the end of a list by entering the new value directly following the null parameter. ICU80 will then prompt for any secondary parameter values that are required.

GENERATE COMMAND

The generate command causes ICU80 to generate the Configuration Object file, Submit file, and Controller Addressable Memory Module (CAM file. The device/file names used are specified in the Description file. The operator must specify which files are to be generated. This is done by responding with a yes or no to prompts for these files.

LIST COMMAND

The list command causes a listing of the description file to be produced on the named ISIS-I1 device/file. If no name is given the listing is directed to the operator's terminal (:CO:).

The format of the list command is as follows:

L(IST) (device/file name)

QUIT COMMAND

This command will return the description file to the state that it was in before ICU80 was initiated. The Quit command is used to exit ICU80 without modifying the description file. The operator is given a message that states that the description file will not be saved and asks if this is acceptable. If the operator enters anything other than YES or Y, ICU80 is not exited and a Command prompt is displayed.

EXIT COMMAND

The Exit command is used to exit ICU80. When the Exit command is executed, it causes the modified deicription file to be saved in the output file specified when ICU80 was initiated. Control is returned to ISIS-I1 after the file has been saved.

COMMENTS

Comments are treated as lines of text. A comment can be inserted on the line preceding any parameter by positioning to the line containing the parameter that the comment is to proceed and entering (ESC) R carriage return. A colon (:) will be inserted on the line preceding the parameter. The operator may now insert a comment or terminate the comment line by entering a carriage return. All comment lines will be preceded by a colon and are limited to 120 characters per line.

The operator can use the Insert comment ((ESC) R) and Delete ((ESC) D) functions to add or delete comments.

The (ESC) R command allows the operator to insert a comment line in the description file preceding the parameter that is being displayed. When the (ESC) R command is entered, a colon (:) will be displayed. This is the beginning of a comment list. The operator may now type in the desired comment. When the comment list is terminated (by entering a carriage return following the colon), the original prompt will again be displayed and the operator may continue.

For example:

```
PARM 1
PARM 2 (ESC) R
:COMMENT
PARM 2
```

This would result in the following:

```
PARM 1
:COMMENT
PARM 2
```

LINKING AND LOCATING THE APPLICATION SYSTEM

After the configuration is defined, the operator must submit the SUBMIT file to cause all the tasks and modules to be linked to the RMX/80 system and the code to be located. The following is an example of a submit operation.

SUBMIT (device/file name)

Where:

file name = the name of the SUBMIT file that was assigned when the description file was created.

APPENDIX A
ICU80 DISK FILES

FILE NAMES

Temporary:

The following files are used by ICU80 as work areas. They will be deleted by ICU80 when Exit or Quit is used to terminate ICU80. Note that if these files exist when ICU80 is initiated they will be destroyed.

ICU801.TMP
ICU802.TMP
ICU803.TMP
ICU804.TMP

Permanent Diskette Files:

These files are distributed with the RMX/80 system and are located on the disk labeled INTERACTIVE CONFIGURATOR for RMX/80. When using ICU80 all of these files must be located on the same diskette.

ICU80
ICU80.DDF
ICU80.OV1
ICU80.OV2
ICU80.OV3
ICU80.OV4
ICU80.OV5
ICU80.MCF

APPENDIX B
SAMPLE CONFIGURATION LISTING

```
:  THIS IS AN EXAMPLE OF USING THE INTERACTIVE CONFIGURATION UTILITY
:  TO CONFIGURE AN RMX/80 APPLICATION SYSTEM.
:  THE KEY CHARACTERISTICS OF THE APPLICATION ARE:
:  RMX/80 TASKS: 1/0 TERMINAL HANDLER, FREE SPACE MANAGER, AND
:  ACTIVE DEBUGGER.
:  USER TASKS: THREE, CALLED USER1, USER2, AND USER3. USER1 SERVICES
:  INTERRUPTS FROM A HIGH-SPEED PERIPHERAL DEVICE; USER2 AND USER3
:  ARE NOT INTERRUPT DRIVEN.
:  EXPLANATIONS OF THE PARAMETER REQUIREMENTS CAN RE ASCERTAINED BY
:  TYPING A QUESTION MARK (?) AFTER THE GIVEN PROMPT.
CONFIG:          :F1:EXAMPL.CNF
SUBMIT:          :F1:EXAMPL.CSD
:  THE CAM MODULE IS NOT USED IF THE SYSTEM DOES NOT INCLUDE DFS.
CAM:          F1:EXAMPL.CAM
LINKORJ:          :F1:SYSTEM.LNK
LINKMAP:          :F1:EXAMPL.LEM
LOCORJ:          SYSTEM.LOC
LOCMAP:          :F1:EXAMPL.LOM
CAMOHJ:          :F1:EXAMPL.LOC
CAMMAP:          :F1:EXAMPL.CLM
BOOT LDR:          NO
LOADABLE:          NO
CPU TYPE:          80/20
:  TO INCLUDE A SYSTEM LEVEL TASK, IT IS ONLY NECESSARY TO ANSWER
:  THE PROMPTS.
:  ICU80 WILL AUTOMATICALLY INCLUDE THE NEEDED EXCHANGES AND TASKS
:  AND MAKE THE CORRECT ENTRIES IN THE INITIAL TASK TABLE (ITT) AND
:  INITIAL EXCHANGE TABLE (IET).
TERM HNDLR:          FULL
  FUNCT:          I/O
  RATE:          1200
  CNTL TBL:          NO
  THISTK:          36
  THIPRI:          112
FSM:          YES
  FSMSTK:          40
  FSMPRI:          130
DEBUG:          ACTIVE
  DRGSTK:          64
  DRGPRI:          140
ANLG HDLR:          NONE
:  TO INCLUDE A USER TASK THE PROMPTS MUST RE ANSWERED. HOWEVER, IF
:  THE TASK HAS NO DEFAULT EXCHANGE, NO VALUE SHOULD RE SUPPLIED
:  FOR THE DEFAULT EXCHANGE.
:  ICU80 WILL AUTOMATICALLY BUILD THE PROPER DATA STRUCTURE IN THE
:  ITT FOR THE USER TASK.
TASK NAME:          USER1
  ENTRY POINT:          USER1
  STK LENGTH:          24
  PRIORITY:          35
  DFLT EXCHG:          RQL2EX
  EXTRA:          0
TASK NAME:          USER2
  ENTRY POINT:          USER2
```

```
  STK LENGTH:          28
  PRIORITY:          135
  DFLT EXCHG:          USXCH2
  EXTRA:          0
TASK NAME:          USER3
  ENTRY POINT:          USER3
  STACK LENGTH:          28
  PRIORITY:          150
  DFLT EXCHG:          USXCH3
  EXTRA:          0
TPSK NAME:          ***
:   ALL USER DEFINED EXCHANGES WHICH ARE TO RE PLACED IN THE IET
:   MUST RE DEFINED HERE. THESE INCLUDE ANY USER DEFINED EXCHANGES
.   WHICH ARE NOT CREATED DYNAMICALLY AT RUN TIME.
EXCHANGE:          USXCH1
  SCOPE:          EXTERNAL
  NTERRUPT:          NO
EXCHANGE          USXCH2
  SCOPE:          EXTERNAL
  INTERRUPT:          NO
E(CHANGE:          USXCH3
  SCOPE.          EXTERNAL
  INTERRUPT:          NO
E XCHANGE:          RESPEX
  SCOPE:          EXTERNAL
  INTERRUPT:          NO
E XCHANGE:          RQL2EX
  SCOPE.          PUBLIC
  INTERRUPT:          YES
E YCHANGE:          ***
DFS:          NO
BASIC:          NO
FORTRAN:          NO
NUCLEUS:          :F1:
EXTENSIONS:          :F1:
:   USRCOD.LIB CONTAINS USER SUPPLIED ORJECT CODE.
LINK:          :F1:USRCOD.LIB
                ***
LINK:
CODE:          0
DATA:          3800H




:   THIS IS AN EXAMPLE OF USING THE INTERACTIVE CONFIGURATION
:   UTILITY TO CONFIGURE AN APPLICATION SYSTEM USING THE DFS
:   SERVICES. THE KEY CHARACTERISTICS OF THIS APPLICATION ARE:
:   DFS SERVICES USED: OPEN, CLOSE, READ, WRITE, AND SEEK.
:   USER TASKS: TWO CALLED UTASK1 AND UTASK2.
:   DISK CONFIGURATION: ONE 201 CONTROLLER WITH TWO DRIVES.
:   TWO 204 CONTROLLERS. THE FIRST IS CONNECTED TO ONE STANDARD SIZE
:   DRIVE, AND THE SECOND IS CONNECTED TO TWO MINT SIZE DRIVES.
:   ONE 206 CONTROLLER WITH THREE DRIVES.
CONFIG:          :F1:DFSCON.OBJ
SUBMIT:          :F1:DFSEXG.CSD
CAM:          :F1:CAMMOD.OBJ
LINKOBJ:          :F1:DFSSYS.LNK
LINKMAP:          :F1:DFSEXG.LEM
LOCOBJ:          :F1:DFSSYS.LOC
```

```
LOCMAP:          :F1:DFSEXG.LOM
CAMORJ:          :F1:CAMMOD
CAMMAP:          :F1:DFSEXG.CLM
ROOT LDR:          NO
LOADABLE:          NO
CPU TYPE:         80/20
TERM HNDLR:            NONE
FSM:          NO
DEBUG:          NONE
ANLG HDLR:           NONE
TASK NAME:           UTASK1
  ENTRY POINT:           UTASK1
  STK LENGTH:          64
  PRIORITY:          150
  DFLT EXCHG:          UTSK1X
  EXTRA:          0
TASK NAME:           UTASK2
  ENTRY POINT:           UTASK2
  STK LENGTH:          80
  PRIORITY:          160
  DFLT EXCHG:          UTSKPX
  EXTRA:          0
TASK NAME:           ***
EXCHANGE:          CNTL1X
  SCOPE:         LOCAL
  INTERRUPT:           NO
EXCHANGE:          CNTL2X
  SCOPE:         LOCAL
  INTERRUPT:           NO
EXCHANGE:          CNTL3X
  SCOPE:         LOCAL
  INTERRUPT:           NO
EXCHANGE:          CNTL4X
  SCOPE:         LOCAL
  INTERRUPT:           NO
EXCHANGE:          UTSK1X
  SCOPE:         LOCAL
  INTERRUPT:           NO
EXCHANGE:          UTSK2X
  SCOPE:         LOCAL
  INTERRUPT:           NO
EXCHANGE:          ***
DFS:          YES
  ATTRIR:          NO
  DELETE:          NO
  FORMAT:          NO
  LOAD:          NO
  RENAME:          NO
  OPEN:          YES
  CLOSE:          YES
  READ:          YES
  WRITE:          YES
    DRSSTK:          48
    DRSPRI:          135
  SEEK:          YES
  DISKIO:          YES
    DIOSTK:          48
    DIOPRI:          129
  CNTRLR:          201
```

```
  ADDRESS:          88H
  LEVEL:            2
  REQUEST:          CNTL1X
  CNTLR STK:        80
  CNTLR PRI:        33
  DRIVE NAME:       SA
    UNIT:           0
  DRIVE NAME:       SB
    UNIT:           1
  DRIVE NAME:       ***
CNTRLR:             204
  ADDRESS:          70H
  LEVEL:            3
  REQUEST:          CNTL2X
  CNTLR STK:        80
  CNTLR PRI:        49
  DRIVE NAME:       SC
    UNIT:           0
    SIZE:           STANDARD
    TYPE:           SA800
    CHIP SELECT:    0
    INDEX           10
      STEP:         8
      SETTLE:       10
      LOAD TIME:    9
  DRIVE NAME:       ***
CNTRLR:             204
  ADDRESS:          60H
  LEVEL:            4
  REQUEST:          CNTL3X
  CNTLR STK:        80
  CNTLR PRI:        66
  DRIVE NAME:       M1
    UNIT:           0
    SIZE:           MINI
    TYPE:           SA400
    CHIP SELECT:    0
    INDEX           10
      STEP:         20
      SETTLE:       10
      LOAD TIME:    10
  DRIVE NAME:       M2
    UNIT:           1
    SIZE:           MINI
    TYPE:           SA400
    CHIP SELECT:    0
    INDEX           10
      STEP:         20
      SETTLE:       10
      LOAD TIME:    10
  DRIVE NAME:       ***
CNTRLR:             206
  ADDRESS:          90H
  LEVEL:            5
  REQUEST:          CNTL4X
  CNTLR STK:        80
CNTLR PRI:          84
  DRIVE NAME:       H1
    UNIT:           0
```

```
    DRIVE NAME:          H2
      UNIT:          1
    DRIVE NAME:          H3
      UNIT:          5
    DRIVE NAME:          ***
  CNTRLR:          ****
  FILES:          4
  BUFFER          ****
BASIC:          NO
FORTRAN:          NO
NUCLEUS:          :F1:
EXTENSIONS:          :F1:
LINK:          :F2:UTASK1.OBJ
LINK:          :F2:UTASK2.OBJ
LINK:          ***
CODE:          0
DATA:          3800H
CAM ADR:          76ADH
```

INDEX

REQUEST FOR READER'S COMMENTS

Intel Corporation attempts to provide documents that meet the needs of all Intel product users. This form lets you participate directly in the documentation process.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this document.

1. Please specify by page any errors you found in this manual.

2. Does the document cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of document for your needs? Is it at the right level? What other types of documents are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this document on a scale of 1 to 10 with 10 being the best rating. _____ – ____

NAME _____ DATE _____

TITLE _____

COMPANY NAME/DEPARTMENT _____

ADDRESS _____

CITY _____ S T A T E ZIP CODE _____

Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1040 SANTACLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
Attn: Technical Publications
3065 Bowers Avenue
Santa Clara, CA 95051